Activity: Hands-on Activity 7.2 Webscraping using BeautifulSoup and Requests

Name: Juliann Vincent B. Quibral

Section: BSCPE22S3

Example of gathering image data using webcam

Open Data and Private Data

1. Open Data The Open Knowledge Foundation describes Open Data as “any content, information or data that people are free to use, reuse, and redistribute without any legal, technological, or social restriction.”
2. Private Data Data related to an expectation of privacy and regulated by a particular country/government

Structured and Unstructured Data

1. Structured Data Data entered and maintained in fixed fields within a file or record Easily entered, classified, queried, and analyzed Relational databases or spreadsheets
2. Unstructured Data Lacks organization Raw data Photo contents, audio, video, web pages, blogs, books, journals, white papers, PowerPoint presentations, articles, email, wikis, word processing documents, and text in general

```
import cv2
from google.colab.patches import cv2_imshow

webcam = cv2.VideoCapture(0)

while True:
    try:
        check, frame = webcam.read()
        if check:  # Check if frame is successfully captured
            print(check)  # prints true as long as the webcam is running
            print(frame)  # prints matrix values of each frame
            cv2_imshow(frame)  # Display the frame
            key = cv2.waitKey(1)

            if key == ord('s'):
                cv2.imwrite(filename='saved_img.jpg', img=frame)
                webcam.release()
                img_new = cv2.imread('saved_img.jpg', cv2.IMREAD_GRAYSCALE)
                img_new = cv2_imshow(img_new)
                cv2.waitKey(1650)
                cv2.destroyAllWindows()
                print("Processing image...")
                img_ = cv2.imread('saved_img.jpg', cv2.IMREAD_ANYCOLOR)
                print("Converting RGB image to grayscale...")
                gray = cv2.cvtColor(img_, cv2.COLOR_BGR2GRAY)
                print("Converted RGB image to grayscale...")
                print("Resizing image to 28x28 scale...")
                img_ = cv2.resize(gray, (28, 28))
                print("Resized...")
                img_resized = cv2.imwrite(filename='saved_img-final.jpg', img=img_)
                print("Image saved!")
                break

            elif key == ord('q'):
                print("Turning off camera.")
                webcam.release()
                print("Camera off.")
                print("Program ended.")
                cv2.destroyAllWindows()
                break
        else:
            print("Unable to capture frame. Check your webcam connection.")
            break

    except KeyboardInterrupt:
        print("Turning off camera.")
        webcam.release()
        print("Camera off.")
        print("Program ended.")
        cv2.destroyAllWindows()
        break

```

```
Unable to capture frame. Check your webcam connection.
```

Example of gathering voice data using microphone

```
!pip3 install sounddevice
```

```
Collecting sounddevice
  Downloading sounddevice-0.4.6-py3-none-any.whl (31 kB)
Requirement already satisfied: CFFI>=1.0 in /usr/local/lib/python3.10/dist-packages (from sounddevice) (1.16.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from CFFI>=1.0->sounddevice) (2.21)
Installing collected packages: sounddevice
Successfully installed sounddevice-0.4.6
```

```
!pip3 install wavio
```

```
Collecting wavio
  Downloading wavio-0.0.8-py3-none-any.whl (9.4 kB)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.10/dist-packages (from wavio) (1.25.2)
Installing collected packages: wavio
Successfully installed wavio-0.0.8
```

```
!pip3 install scipy
```

```
Requirement already satisfied: scipy in /usr/local/lib/python3.10/dist-packages (1.11.4)
Requirement already satisfied: numpy<1.28.0,>=1.21.6 in /usr/local/lib/python3.10/dist-packages (from scipy) (1.25.2)
```

```
!apt-get install libportaudio2
```

```
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following NEW packages will be installed:
  libportaudio2
0 upgraded, 1 newly installed, 0 to remove and 39 not upgraded.
Need to get 65.3 kB of archives.
After this operation, 223 kB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy/universe amd64 libportaudio2 amd64 19.6.0-1.1 [65.3 kB]
Fetched 65.3 kB in 1s (81.0 kB/s)
Selecting previously unselected package libportaudio2:amd64.
(Reading database ... 121753 files and directories currently installed.)
Preparing to unpack .../libportaudio2_19.6.0-1.1_amd64.deb ...
Unpacking libportaudio2:amd64 (19.6.0-1.1) ...
Setting up libportaudio2:amd64 (19.6.0-1.1) ...
Processing triggers for libc-bin (2.35-0ubuntu3.4) ...
/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_5.so.3 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbb.so.12 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbmalloc_proxy.so.2 is not a symbolic link

/sbin/ldconfig.real: /usr/local/lib/libtbbbind_2_0.so.3 is not a symbolic link
```

```
!pip install sounddevice --upgrade
```

```
Requirement already satisfied: sounddevice in /usr/local/lib/python3.10/dist-packages (0.4.6)
Requirement already satisfied: CFFI>=1.0 in /usr/local/lib/python3.10/dist-packages (from sounddevice) (1.16.0)
Requirement already satisfied: pycparser in /usr/local/lib/python3.10/dist-packages (from CFFI>=1.0->sounddevice) (2.21)
```

```
 1 # import required libraries
 2 import sounddevice as sd
 3 from scipy.io.wavfile import write
 4 import wavio as wv
 5 
 6 # Sampling frequency
 7 freq = 48000
 8 
 9 # Recording duration
10 duration = 5
11 
12 # Start recorder with the given values
13 # of duration and sample frequency
14 recording = sd.rec(int(duration * freq),
15  samplerate=freq, channels=2)
16 
17 # Record audio for the given number of seconds
18 sd.wait()
19 
20 # This will convert the NumPy array to an audio
21 # file with the given sampling frequency
22 write("recording0.wav", freq, recording)
23 # Convert the NumPy array to audio file
24 wv.write("recording1.wav", recording, freq, sampwidth=2)
```

```
---------------------------------------------------------------------------
PortAudioError                            Traceback (most recent call last)
<ipython-input-21-3f46ebebbb4e> in <cell line: 14>()
     12 # Start recorder with the given values
     13 # of duration and sample frequency
---> 14 recording = sd.rec(int(duration * freq),
     15  samplerate=freq, channels=2)
     16 

                          5 frames

/usr/local/lib/python3.10/dist-packages/sounddevice.py in query_devices(device, kind)
    567     info = _lib.Pa_GetDeviceInfo(device)
    568     if not info:
--> 569         raise PortAudioError(f'Error querying device {device}')
    570     assert info.structVersion == 2
    571     name_bytes = _ffi.string(info.name)

PortAudioError: Error querying device -1
```

Web Scraping

```
!pip install bs4
```

```
Collecting bs4
  Downloading bs4-0.0.2-py2.py3-none-any.whl (1.2 kB)
Requirement already satisfied: beautifulsoup4 in /usr/local/lib/python3.10/dist-packages (from bs4) (4.12.3)
Requirement already satisfied: soupsieve>1.2 in /usr/local/lib/python3.10/dist-packages (from beautifulsoup4->bs4) (2.5)
Installing collected packages: bs4
Successfully installed bs4-0.0.2
```

```
pip install requests
```

```
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (2.31.0)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests) (3.6)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.10/dist-packages (from requests) (2.0.7)
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.10/dist-packages (from requests) (2024.2.2)
```

```
import requests
from bs4 import BeautifulSoup

def getdata(url):
 r = requests.get(url)
 return r.text

htmldata = getdata("https://www.google.com/")
soup = BeautifulSoup(htmldata, 'html.parser')
for item in soup.find_all('img'):
 print(item['src'])

```

```
/images/branding/googlelogo/1x/googlelogo_white_background_color_272x92dp.png
```

```
pip install selenium
```

```
Collecting selenium
  Downloading selenium-4.18.1-py3-none-any.whl (10.0 MB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 10.0/10.0 MB 52.9 MB/s eta 0:00:00
Requirement already satisfied: urllib3[socks]<3,>=1.26 in /usr/local/lib/python3.10/dist-packages (from selenium) (2.0.7)
Collecting trio~=0.17 (from selenium)
  Downloading trio-0.25.0-py3-none-any.whl (467 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 467.2/467.2 kB 40.4 MB/s eta 0:00:00
Collecting trio-websocket~=0.9 (from selenium)
  Downloading trio_websocket-0.11.1-py3-none-any.whl (17 kB)
Requirement already satisfied: certifi>=2021.10.8 in /usr/local/lib/python3.10/dist-packages (from selenium) (2024.2.2)
Requirement already satisfied: typing_extensions>=4.9.0 in /usr/local/lib/python3.10/dist-packages (from selenium) (4.10.0)
Requirement already satisfied: attrs>=23.2.0 in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (23.2.0)
Requirement already satisfied: sortedcontainers in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (2.4.0)
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (3.6)
Collecting outcome (from trio~=0.17->selenium)
  Downloading outcome-1.3.0.post0-py2.py3-none-any.whl (10 kB)
Requirement already satisfied: sniffio>=1.3.0 in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (1.3.1)
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages (from trio~=0.17->selenium) (1.2.0)
Collecting wsproto>=0.14 (from trio-websocket~=0.9->selenium)
  Downloading wsproto-1.2.0-py3-none-any.whl (24 kB)
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/python3.10/dist-packages (from urllib3[socks]<3,>=1.26->sel
Collecting h11<1,>=0.9.0 (from wsproto>=0.14->trio-websocket~=0.9->selenium)
  Downloading h11-0.14.0-py3-none-any.whl (58 kB)
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 58.3/58.3 kB 8.2 MB/s eta 0:00:00
Installing collected packages: outcome, h11, wsproto, trio, trio-websocket, selenium
Successfully installed h11-0.14.0 outcome-1.3.0.post0 selenium-4.18.1 trio-0.25.0 trio-websocket-0.11.1 wsproto-1.2.0
```

Image Scraping using Selenium

```
!pip install selenium
!apt-get update # to update ubuntu to correctly run apt install
!apt install chromium-chromedriver
!cp /usr/lib/chromium-browser/chromedriver /usr/bin
import sys
sys.path.insert(0,'/usr/lib/chromium-browser/chromedriver')

from selenium import webdriver
import time
import requests
import shutil
import os
import getpass
import urllib.request
import io
import time
from PIL import Image
user = getpass.getuser()
chrome_options = webdriver.ChromeOptions()
chrome_options.add_argument('--headless')
chrome_options.add_argument('--no-sandbox')
chrome_options.add_argument('--disable-dev-shm-usage')
driver = webdriver.Chrome('chromedriver',chrome_options=chrome_options)

search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&ved=0CAIQpwVqFwoTCKCa1c6s4-oCFQAAAAAdAAAAABAC&biw=1251&bih=
driver.get(search_url.format(q='Car'))

def scroll_to_end(driver):
  driver.execute_script("window.scrollTo(0, document.body.scrollHeight);")
  time.sleep(5)#sleep_between_interactions

def getImageUrls(name,totalImgs,driver):
  search_url = "https://www.google.com/search?q={q}&tbm=isch&tbs=sur%3Afc&hl=en&ved=0CAIQpwVqFwoTCKCa1c6s4-oCFQAAAAAdAAAAABAC&biw=1251&bi
  driver.get(search_url.format(q=name))
  img_urls = set()
  img_count = 0
  results_start = 0

  while(img_count<totalImgs): #Extract actual images now

    scroll_to_end(driver)

    thumbnail_results = driver.find_elements_by_xpath("//img[contains(@class,'Q4LuWd')]")
    totalResults=len(thumbnail_results)
    print(f"Found: {totalResults} search results. Extracting links from{results_start}:{totalResults}")

    for img in thumbnail_results[results_start:totalResults]:

      img.click()
      time.sleep(2)
      actual_images = driver.find_elements_by_css_selector('img.n3VNCb')
      for actual_image in actual_images:
        if actual_image.get_attribute('src') and 'https' in actual_image.get_attribute('src'):
          img_urls.add(actual_image.get_attribute('src'))

      img_count=len(img_urls)

      if img_count >= totalImgs:
        print(f"Found: {img_count} image links")
        break
      else:
        print("Found:", img_count, "looking for more image links ...")
        load_more_button = driver.find_element_by_css_selector(".mye4qd")
        driver.execute_script("document.querySelector('.mye4qd').click();")
        results_start = len(thumbnail_results)
  return img_urls

def downloadImages(folder_path,file_name,url):
    try:
      image_content = requests.get(url).content
    except Exception as e:
      print(f"ERROR - COULD NOT DOWNLOAD {url} - {e}")

    try:
      image_file = io.BytesIO(image_content)
      image = Image.open(image_file).convert('RGB')

      file_path = os.path.join(folder_path, file_name)

      with open(file_path, 'wb') as f:
        image.save(f, "JPEG", quality=85)
      print(f"SAVED - {url} - AT: {file_path}")
    except Exception as e:
      print(f"ERROR - COULD NOT SAVE {url} - {e}")

def saveInDestFolder(searchNames,destDir,totalImgs,driver):
    for name in list(searchNames):
      path=os.path.join(destDir,name)
      if not os.path.isdir(path):
        os.mkdir(path)
      print('Current Path',path)
      totalLinks=getImageUrls(name,totalImgs,driver)
      print('totalLinks',totalLinks)
    if totalLinks is None:
        print('images not found for :',name)

    else:
      for i, link in enumerate(totalLinks):
        file_name = f"{i:150}.jpg"
        downloadImages(path,file_name,link)

searchNames=['cat']
destDir=f'/content/drive/My Drive/Colab Notebooks/Dataset/'
totalImgs=5

saveInDestFolder(searchNames,destDir,totalImgs,driver)
```

```
Requirement already satisfied: selenium in /usr/local/lib/python3.10/dist-packages (4.18
Requirement already satisfied: urllib3[socks]<3,>=1.26 in /usr/local/lib/python3.10/dist
Requirement already satisfied: trio~=0.17 in /usr/local/lib/python3.10/dist-packages (fr
Requirement already satisfied: trio-websocket~=0.9 in /usr/local/lib/python3.10/dist-pac
Requirement already satisfied: certifi>=2021.10.8 in /usr/local/lib/python3.10/dist-pack
Requirement already satisfied: typing_extensions>=4.9.0 in /usr/local/lib/python3.10/dis
Requirement already satisfied: attrs>=23.2.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: sortedcontainers in /usr/local/lib/python3.10/dist-packag
Requirement already satisfied: idna in /usr/local/lib/python3.10/dist-packages (from tri
Requirement already satisfied: outcome in /usr/local/lib/python3.10/dist-packages (from
Requirement already satisfied: sniffio>=1.3.0 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: exceptiongroup in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: wsproto>=0.14 in /usr/local/lib/python3.10/dist-packages
Requirement already satisfied: pysocks!=1.5.7,<2.0,>=1.5.6 in /usr/local/lib/python3.10/
Requirement already satisfied: h11<1,>=0.9.0 in /usr/local/lib/python3.10/dist-packages
Get:1 https://cloud.r-project.org/bin/linux/ubuntu jammy-cran40/ InRelease [3,626 B]
Get:2 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  InRele
Get:3 http://security.ubuntu.com/ubuntu jammy-security InRelease [110 kB]
Hit:4 http://archive.ubuntu.com/ubuntu jammy InRelease
Get:5 https://developer.download.nvidia.com/compute/cuda/repos/ubuntu2204/x86_64  Packag
Get:6 http://archive.ubuntu.com/ubuntu jammy-updates InRelease [119 kB]
Hit:7 http://archive.ubuntu.com/ubuntu jammy-backports InRelease
Hit:8 https://ppa.launchpadcontent.net/c2d4u.team/c2d4u4.0+/ubuntu jammy InRelease
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 Packages [1,356 kB]
Hit:10 https://ppa.launchpadcontent.net/deadsnakes/ppa/ubuntu jammy InRelease
Hit:11 https://ppa.launchpadcontent.net/graphics-drivers/ppa/ubuntu jammy InRelease
Get:12 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 Packages [1,898 kB]
Hit:13 https://ppa.launchpadcontent.net/ubuntugis/ppa/ubuntu jammy InRelease
Fetched 4,261 kB in 2s (2,625 kB/s)
Reading package lists... Done
Reading package lists... Done
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  apparmor chromium-browser libfuse3-3 liblzo2-2 libudev1 snapd squashfs-tools systemd-h
  udev
Suggested packages:
  apparmor-profiles-extra apparmor-utils fuse3 zenity | kdialog
The following NEW packages will be installed:
  apparmor chromium-browser chromium-chromedriver libfuse3-3 liblzo2-2 snapd squashfs-to
  systemd-hwe-hwdb udev
The following packages will be upgraded:
  libudev1
1 upgraded, 9 newly installed, 0 to remove and 38 not upgraded.
Need to get 26.4 MB of archives.
After this operation, 116 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 apparmor amd64 3.0.4-2ub
Get:2 http://archive.ubuntu.com/ubuntu jammy/main amd64 liblzo2-2 amd64 2.10-2build3 [53
Get:3 http://archive.ubuntu.com/ubuntu jammy/main amd64 squashfs-tools amd64 1:4.5-3buil
Get:4 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 libudev1 amd64 249.11-0u
Get:5 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 udev amd64 249.11-0ubunt
Get:6 http://archive.ubuntu.com/ubuntu jammy/main amd64 libfuse3-3 amd64 3.10.5-1build1
Get:7 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 snapd amd64 2.58+22.04.1
Get:8 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 chromium-browser amd
Get:9 http://archive.ubuntu.com/ubuntu jammy-updates/universe amd64 chromium-chromedrive
Get:10 http://archive.ubuntu.com/ubuntu jammy-updates/main amd64 systemd-hwe-hwdb all 24
Fetched 26.4 MB in 1s (18.6 MB/s)
```

Web Scraping of Movies Information using BeautifulSoup

```
from requests import get
url = 'https://hurawatch.cc/search/logan'
response = get(url)
print(response.text[:500])
```

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head>
    <title>Search results for &#39;logan&#39; Movies &amp; Tv Series Hurawatch</title>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8"/>

    <meta name="robots" content="index, follow">
    <meta name="revisit-after" content="1 days">

<meta http-equiv="content-language" content="en"/>
<link rel="dns-prefetch" href="//www.google-analytics.com">
<link rel="dns-prefetch" href="//www.gstat
```

```
from bs4 import BeautifulSoup
html_soup = BeautifulSoup(response.text, 'html.parser')
headers = {'Accept-Language': 'en-US,en;q=0.8'}
type(html_soup)

```

```
bs4.BeautifulSoup
def __call__(*args, **kwargs)

A data structure representing a parsed HTML or XML document.

Most of the methods you'll call on a BeautifulSoup object are inherited from
PageElement or Tag.

Internally, this class defines the basic interface called by the
```

```
movie_containers = html_soup.find_all('div', class_ = 'flw-item')
print(type(movie_containers))
print(len(movie_containers))
```

```
<class 'bs4.element.ResultSet'>
6
```

```
first_movie = movie_containers[0]
first_movie
```

```
<div class="flw-item">
<div class="film-poster">
<div class="pick film-poster-quality">HD</div>
<img alt="Logan" class="film-poster-img lazyload" data-src="https://img.hurawatch.cc/xxrz/250x400/348/16/f7/16f7b48a3df281f25cb746394488ea9d/16f7b48a3df281f25cb746394488ea9d.jpg"
title="Logan"/>
<a class="film-poster-ahref flw-item-tip" href="/movie/watch-logan-online-19754" title="Logan"><i class="fa fa-play"></i></a>
</div>
<div class="film-detail">
<h2 class="film-name"><a href="/movie/watch-logan-online-19754" title="Logan">Logan</a>
</h2>
<div class="fd-infor">
<span class="fdi-item">2017</span>
<span class="dot"></span>
<span class="fdi-item fdi-duration">137m</span>
<span class="float-right fdi-type">Movie</span>
</div>
<div class="clearfix"></div>
</div>
<div class="clearfix"></div>
</div>
```

```
first_movie.div
```

```
<div class="film-poster">
<div class="pick film-poster-quality">HD</div>
<img alt="Logan" class="film-poster-img lazyload" data-src="https://img.hurawatch.cc/xxrz/250x400/348/16/f7/16f7b48a3df281f25cb746394488ea9d/16f7b48a3df281f25cb746394488ea9d.jpg"
title="Logan"/>
<a class="film-poster-ahref flw-item-tip" href="/movie/watch-logan-online-19754" title="Logan"><i class="fa fa-play"></i></a>
</div>
```

```
first_movie.a
```

```
<a class="film-poster-ahref flw-item-tip" href="/movie/watch-logan-online-19754" title="Logan"><i class="fa fa-play"></i></a>
```

```
first_movie.h2
```

```
<h2 class="film-name"><a href="/movie/watch-logan-online-19754" title="Logan">Logan</a>
</h2>
```

```
first_movie.h2.a
```

```
<a href="/movie/watch-logan-online-19754" title="Logan">Logan</a>
```

```
first_name = first_movie.h2.a.text
first_name
```

```
'Logan'
```

```
first_year = first_movie.find('span', class_='fdi-item')
if first_year:
    print(first_year.text)
else:
    print("Year information not found")
```

```
2017
```

```
first_year = first_year.text
first_year
```

```
'2017'
```

Rating doesn't exist in my link

```
first_movie.strong
```

```
first_imdb = float(first_movie.strong.text)
first_imdb
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-88-92faeb51c9f2> in <cell line: 1>()
----> 1 first_imdb = float(first_movie.strong.text)
      2 first_imdb

AttributeError: 'NoneType' object has no attribute 'text'
```

Metascore doesn't exist in my link

```
first_mscore = first_movie.find('span', class_ = 'metascore favorable')
first_mscore = int(first_mscore.text)
print(first_mscore)
```

```
---------------------------------------------------------------------------
AttributeError                            Traceback (most recent call last)
<ipython-input-89-889bc009bd72> in <cell line: 2>()
      1 first_mscore = first_movie.find('span', class_ = 'metascore favorable')
----> 2 first_mscore = int(first_mscore.text)
      3 print(first_mscore)

AttributeError: 'NoneType' object has no attribute 'text'
```

Votes doesn't exist in my link

```
first_votes = first_movie.find('span', attrs = {'name':'nv'})
first_votes
```

```
first_votes['data-value']
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-92-2d836d02a09a> in <cell line: 1>()
----> 1 first_votes['data-value']

TypeError: 'NoneType' object is not subscriptable
```

```
first_votes = int(first_votes['data-value'])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-93-e337b21fe258> in <cell line: 1>()
----> 1 first_votes = int(first_votes['data-value'])

TypeError: 'NoneType' object is not subscriptable
```

However the site have a time duration

```
first_duration = first_movie.find('span', class_='fdi-item fdi-duration')
if first_duration:
    print(first_duration.text)
else:
    print("Duration information not found")

```

```
137m
```

```
# Lists to store the scraped data in
names = []
years = []
durations = []


for container in movie_containers:
    if container.find('div', class_='fd-infor') is not None:
        # Name
        name = container.h2.a.text
        names.append(name)

        # Year
        year = container.find('span', class_='fdi-item').text
        years.append(year)

        # Duration
        duration_element = container.find('span', class_='fdi-item fdi-duration')
        if duration_element is not None:
            duration = duration_element.text
        else:
            duration = 'Not available'
        durations.append(duration)

print(names)
print(years)
print(durations)
```

```
['Logan', 'Logan Lucky', 'The Night Logan Woke Up', 'The Taking of Deborah Logan', 'The Two Worlds of Jennie Logan', "Logan's Run"]
['2017', '2017', 'SS 1', '2014', '1979', '1976']
['137m', '119m', 'Not available', '90m', '94m', '119m']
```

```
import pandas as pd
test_df = pd.DataFrame({
'movie': names,
'year': years,
'duration': duration,
})
print(test_df.info())
test_df
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   movie     6 non-null      object
 1   year      6 non-null      object
 2   duration  6 non-null      object
dtypes: object(3)
memory usage: 272.0+ bytes
None
                                movie  year  duration
 0                              Logan  2017      119m
 1                        Logan Lucky  2017      119m
 2              The Night Logan Woke Up  SS 1      119m
 3          The Taking of Deborah Logan  2014      119m
 4     The Two Worlds of Jennie Logan  1979      119m
 5                        Logan's Run  1976      119m
```

My website doesn't have another batch of movies and is limited to just 6/6

```
from time import time, sleep
from random import randint
from IPython.core.display import clear_output
from requests import get
from bs4 import BeautifulSoup
from warnings import warn

# Lists to store the scraped data
names = []
years = []
durations = []  # Adding duration to store movie durations
# IMDb ratings, Metascores, and votes are already present in the code

# Preparing the monitoring of the loop
start_time = time()
requests = 0

# Pages and years_url as per your previous code
pages = ['1', '2', '3', '4', '5']
years_url = ['2017', '2018', '2019', '2020']

for year_url in years_url:
    for page in pages:
        # Make a get request
        response = get('https://hurawatch.cc/search/logan' + year_url +
                       '&sort=num_votes,desc&page=' + page)

        # Pause the loop
        sleep(randint(8, 15))

        # Monitor the requests
        requests += 1
        elapsed_time = time() - start_time
        print('Request:{}; Frequency: {} requests/s'.format(requests, requests / elapsed_time))
        clear_output(wait=True)

        # Throw a warning for non-200 status codes
        if response.status_code != 200:
            warn('Request: {}; Status code: {}'.format(requests, response.status_code))

        # Parse the content of the request with BeautifulSoup
        page_html = BeautifulSoup(response.text, 'html.parser')

        # Select all the movie containers from a single page
        mv_containers = page_html.find_all('div', class_='fdi-item')

        # For every movie in the containers
        for container in mv_containers:
            # Extracting movie name
            name = container.h2.a.text
            names.append(name)

            # Extracting movie year
            year = container.h2.find('span', class_='fdi-item').text
            years.append(year)

            # Extracting movie duration
            duration = container.find('span', class_='fdi-item fdi-duration').text
            durations.append(duration)

# Check the scraped data
print(names)
print(years)
print(durations)

```

```
---------------------------------------------------------------------------
KeyboardInterrupt                         Traceback (most recent call last)
<ipython-input-146-c6e20e626e83> in <cell line: 22>()
     23     for page in pages:
     24         # Make a get request
---> 25         response = get('https://hurawatch.cc/search/logan' + year_url +
     26                        '&sort=num_votes,desc&page=' + page)
     27

                                 13 frames
/usr/lib/python3.10/ssl.py in read(self, len, buffer)
   1157         try:
   1158             if buffer is not None:
-> 1159                 return self._sslobj.read(len, buffer)
   1160             else:
   1161                 return self._sslobj.read(len)

KeyboardInterrupt:
```

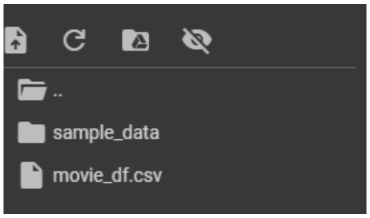
```
movie_ratings = pd.DataFrame({
'movie': names,
'year': years,
'duration': duration,
})
print(movie_ratings.info)
movie_ratings.head(10)
```

```
import pandas as pd
movie_df = pd.DataFrame({
'movie': names,
'year': years,
'duration': duration,
})
print(movie_df.info())
movie_df
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6 entries, 0 to 5
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype 
---  ------    --------------  ----- 
 0   movie     6 non-null      object
 1   year      6 non-null      object
 2   duration  6 non-null      object
dtypes: object(3)
memory usage: 272.0+ bytes
None
```

	movie	year	duration
0	Logan	2017	119m
1	Logan Lucky	2017	119m
2	The Night Logan Woke Up	SS 1	119m
3	The Taking of Deborah Logan	2014	119m
4	The Two Worlds of Jennie Logan	1979	119m
5	Logan's Run	1976	119m

```
movie_df.to_csv('/content/movie_df.csv')
```



Data preparation

```
movie_df['year'].unique()
```

```
array(['2017', 'SS 1', '2014', '1979', '1976'], dtype=object)
```